論文 / 著書情報
Article / Book Information

Title	Certificateless Aggregate Signature Schemes with Improved Security
Authors	Nguyen Quoc Viet, Wakaha Ogata
Citation	IEICE Trans. on Fundamentals, Vol. E98-A, No. 1, pp. 92-99
Pub. date	2015, 1
URL	http://search.ieice.org/
Copyright	(c) 2015 Institute of Electronics, Information and Communication Engineers

IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E98-A NO. 1
JANUARY 2015

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. FUNDAMENTALS, VOL.E98–A, NO.1 JANUARY 2015

PAPER Special Section on Cryptography and Information Security

Certificateless Aggregate Signature Schemes with Improved Security

Nguyen Quoc VIET[†], *Nonmember and* **Wakaha OGATA**[†a)], *Member*

SUMMARY A certificateless aggregate signature scheme saves cost from complicated certificate management in PKI and compresses many signatures on different messages signed by different users to one single signature. It is originally required to be secure against a conspiring group of malicious signers (*type I adversary*) and against malicious KGC (*type II adversary*). In this paper, we define a novel fundamental type of adversary for certificateless aggregate signature schemes, type III adversary, called *malicious KGC & Signers Coalition*, who can break Zhang-Zhang scheme. We also propose two new certificateless aggregate schemes which are provably secure against all three types of adversary.

key words: *certificateless signature, aggregate signature, coalition attack*

1. Introduction

A digital signature must be as short as possible to save communication bandwidth and storage. Since 2003, soon after Al-Riyami presented the model of Certificateless Public Key Cryptography(CL-PKC) in [1] and Boneh introduced the first Aggregate Signature (AS) scheme in [2], researchers have been working on drawing an efficient combination of CL-PKC and AS, presently known as Certificateless Aggregate Signature scheme (CLAS). The main motivation of aggregate signature scheme is compactness: an aggregate signature is a single short string that convinces any verifier that n signers $S_1, \ldots, S_n$ indeed signed n messages $M_1, \ldots, M_n$ respectively. Gentry introduced Identity-Based aggregate signatures in [3] by combining identity-based cryptography and aggregate signature scheme; however, it faces with the inherent key escrow problem in identity-based cryptosystems. In [4], Gorantla and Saxena built an efficient certificateless signature scheme which exploits the significant advantages of certificateless cryptosystem: avoidance of the usage of certificates and not suffering from key escrow. As a unification of the advantages of CL-PKC and the compactness of AS, the first two CLAS schemes and their security model had been proposed in [5] by Gong et al. Afterwards, in [6] Zhang and Zhang pointed out unreasonable restrictions that were imposed on adversary model in [5]. They proposed a new security definition and a new scheme, Zhang-Zhang scheme, which is secure under the new definition and efficient in the sign phase, yet lost in aggregate signature's length. In [7], Zhang et al. introduced another scheme, Zhang-Qin-Wu-Zhang scheme, with a major improvement in aggregate signature's length. Later, Shim presented coalition attacks [8] on Zhang-Zhang scheme to show that a KGC conspiring with a malicious signer can forge a certificateless aggregate signature on any set of messages for any group that includes the malicious user. Although such type of attacks has not been assumed in CLASs, Wong et al. [9] already defined similar attack, so-called "malicious-but passive KGC" attack, in (non-aggregate) certificateless signature.

In this paper, we first modify Wong et al.'s "malicious-but passive KGC" attack to fit CLAS, we then propose two new efficient CLAS schemes that are secure in the new security model.

Manuscript received March 18, 2014.
Manuscript revised July 15, 2014.

[†]The authors are with Tokyo Institute of Technology, Tokyo, 152-8552 Japan.
a) E-mail: wakaha@mot.titech.ac.jp

DOI: 10.1587/transfun.E98.A.92

2. Preliminaries

2.1 Model of CLAS Schemes

A Key Generation Center (KGC), multiple users whose identities are $ID_1, ID_2, \ldots$ together participate in a CLAS scheme. The scheme basically consists of the following six algorithms:

Setup: KGC runs this algorithm which accepts security parameter 1^ℓ to generate a master-key mk and a list of system parameters $params$. In the following algorithms, $params$ is commonly used as input, and we will omit it.

PartialPrivateKeyExtr: This algorithm is also performed by KGC. It takes a user's identity ID_i and a master-key mk as input, and generates the user's partial private key D_i.

UserKeyGen: A user runs this algorithm by inputting his identity ID_i. It outputs the user's secret key x_i and his public key P_i. The user's signing key consists of his partial private key D_i and his secret key x_i.

Sign: For each user, this algorithm takes his identity ID_i, his public key P_i, his signing key (D_i, x_i), a message M_i, and a state information Δ as inputs. It generates a valid signature σ_i on message M_i under identity ID_i and public key P_i. State information Δ is used to control which signatures can be aggregated with, i.e., only the signatures that were generated with the same Δ can be aggregated.

Aggr: Under receiving inputs which are a list of identities $(ID_1, \ldots, ID_n)$ and their corresponding public keys $(P_1, \ldots, P_n)$, signatures $(\sigma_1, \ldots, \sigma_n)$ of messages

$(M_1, \ldots, M_n)$, and a common state information Δ, this algorithm outputs an aggregate signature σ.

Verify: The verifier runs this algorithm by taking identities $(ID_1, \ldots, ID_n)$, public keys $(P_1, \ldots, P_n)$, messages $(M_1, \ldots, M_n)$, an aggregate signature σ, and a state information Δ as inputs. It outputs 1 (accept) or 0 (reject).

For correctness,

$$\text{Verify}((ID_1, \ldots, ID_n), (P_1, \ldots, P_n), (M_1, \ldots, M_n), \sigma, \Delta)$$

must be 1 if

$$\begin{aligned}
&D_i \leftarrow \text{PartialPrivateKeyExtr}(ID_i, mk),\\
&(x_i, P_i) \leftarrow \text{UserKeyGen}(ID_i),\\
&\sigma_i \leftarrow \text{Sign}(ID_i, P_i, (D_i, x_i), M_i, \Delta),\\
&\sigma \leftarrow \text{Aggr}((ID_1, \ldots, ID_n), (P_1, \ldots, P_n),\\
&\qquad (M_1, \ldots, M_n), (\sigma_1, \ldots, \sigma_n), \Delta)
\end{aligned}$$

for all i.

2.2 Security Definitions

Al-Riyami and Paterson defined in [1] two types of adversaries for certificateless public key cryptography: Type I adversary and Type II adversary. Type I adversary represents a group of malicious users; it can replace the public key of any user with a value of its choice but it cannot have access to the master-key. Reversely, Type II adversary has access to the master-key but cannot replace any public keys because it acts a malicious KGC.

Two following games between a challenger $\mathscr{C}$ and adversaries $\mathscr{A}_1$ and $\mathscr{A}_2$ model in details the characteristics of two types of adversaries described above.

Game 1 for Type I adversary.

Setup: $\mathscr{C}$ runs Setup algorithm, obtains a master-key mk and a system parameter $params$. $\mathscr{C}$ sends $params$ to $\mathscr{A}_1$ while keeps mk in secret.

Attack: Adversary $\mathscr{A}_1$ can request the following types of queries in a polynomial number of time:

- Partial-Private-Key query $\mathcal{PPK}(ID_i)$: $\mathscr{A}_1$ can request the partial private key of any user. In respond, $\mathscr{C}$ returns the partial private key D_i = PartialPrivateKeyExtr(ID_i, mk).
- Public-Key query $\mathcal{PK}(ID_i)$: To respond $\mathscr{A}_1$'s request for a public key of a user whose identity is ID_i, $\mathscr{C}$ runs UserKeyGen(ID_i) to obtain (x_i, P_i), and returns the public key P_i. $\mathscr{C}$ stores (ID_i, x_i, P_i).
- Public-Key-Replacement query $\mathcal{PKR}(ID_i, P_i')$: $\mathscr{A}_1$ sends these queries to replace the public key P_i of user identified as ID_i with its selected value P_i'. $\mathscr{C}$ updates data for identity ID_i from (ID_i, x_i, P_i) to $(ID_i, \perp, P_i')$.
- Secret-Key query $\mathcal{SK}(ID_i)$: This type of adversary can request the secret key of any user. In return, $\mathscr{C}$ outputs the secret key x_i for identity ID_i[†].
- Sign query $\mathcal{S}(ID_i, P_i, M_i, \Delta_i)$: On receiving this query, $\mathscr{C}$ runs Sign$(ID_i, P_i, (D_i, x_i), M_i, \Delta)$ to obtain a valid signature σ_i, and sends σ_i to the adversary.

Forgery: $\mathscr{A}_1$ outputs a list of identities $(ID_1^*, \ldots, ID_n^*)$ and corresponding public keys $(P_1^*, \ldots, P_n^*)$, messages $(M_1^*, \ldots, M_n^*)$, a state information Δ^* and an aggregate signature σ^*.

$\mathscr{A}_1$ wins Game 1 if:

1. Verify$((ID_1^*, \ldots, ID_n^*), (P_1^*, \ldots, P_n^*), (M_1^*, \ldots, M_n^*), \sigma^*, \Delta^*) = 1$ holds, and
2. there exists at least one $i \in \{1, \ldots, n\}$ such that $\mathscr{A}_1$ did not request ID_i^*'s partial private key, and did not make sign query $\mathcal{S}(ID_i^*, P_i^*, M_i^*, \Delta^*)$.

If the probability that $\mathscr{A}_1$ wins Game 1 is negligible for any polynomial time bounded adversary $\mathscr{A}_1$, we say that the scheme is secure against type I attacks.

Game 2 for Type II adversary.

Setup: $\mathscr{C}$ runs Setup algorithm to obtain mk and $params$. Unlike to Type I adversary, $\mathscr{C}$ sends both mk and $params$ to $\mathscr{A}_2$.

Attack: Adversary $\mathscr{A}_2$ can request Public-Key query $\mathcal{PK}(ID_i)$, Secret-Key query $\mathcal{SK}(ID_i)$, and Sign query $\mathcal{S}(ID_i, P_i, M_i, \Delta_i)$ in polynomial number of times. $\mathscr{C}$ responds these queries in the same way as in Game 1. Note that $\mathscr{A}_2$ does not allow to issue Public-Key-Replacement query, and it does not need to issue Partial-Private-Key query since it knows mk.

Forgery: Finally, $\mathscr{A}_2$ outputs $(ID_1^*, \ldots, ID_n^*)$, $(P_1^*, \ldots, P_n^*)$, $(M_1^*, \ldots, M_n^*)$, Δ^*, and σ^*.

$\mathscr{A}_2$ wins Game 2 if:

1. Verify$((ID_1^*, \ldots, ID_n^*), (P_1^*, \ldots, P_n^*), (M_1^*, \ldots, M_n^*), \sigma^*, \Delta^*) = 1$ holds, and
2. there exists at least one $i \in \{1, \ldots, n\}$ such that $\mathscr{A}_2$ did not make Sign query $\mathcal{S}(ID_i^*, P_i^*, M_i^*, \Delta^*)$ and Secret-Key query $\mathcal{SK}(ID_i^*)$.

If the probability that $\mathscr{A}_2$ wins Game 2 is negligible for any polynomial time bounded adversaries $\mathscr{A}_2$, we say that the scheme is secure against type II attacks.

2.3 Bilinear Maps

Many available certificateless schemes such as [3], [5]–[7], [10] and our schemes use a bilinear map, which is often called a "pairing." Throughout this paper, q is a ℓ-bit prime number, $\mathbb{G}_1$ denotes an additive group of order q and $\mathbb{G}_2$ is a multiplicative group of the same order. P is a generator of $\mathbb{G}_1$. A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties:

[†]If the secret key for ID_i has not be generated, $\mathscr{C}$ makes Public-Key query $\mathcal{PK}(ID_i)$ internally.

- Bilinearity: given $Q, W, Z \in \mathbb{G}_1$, we have

$$\hat{e}(Q, W + Z) = \hat{e}(Q, W) \times \hat{e}(Q, Z)$$

and

$$\hat{e}(Q + W, Z) = \hat{e}(Q, Z) \times \hat{e}(W, Z).$$

- Non-degenerate: $\hat{e}(P, P) \neq 1_{G_2}$.
- The map $\hat{e}(\cdot, \cdot)$ is efficiently computable.

The computational Diffie-Hellman (CDH) problem in $\mathbb{G}_1$ helps to form the basis of security in many cryptographic schemes: given a generator P of $\mathbb{G}_1$, the group order q, and given (aP, bP) for randomly chosen $a, b \in \mathbb{Z}_q^*$, compute abP.

We assume the existence of an algorithm that, taking a security parameter 1^ℓ, generates $(\mathbb{G}_1, \mathbb{G}_2, q, P, e)$ such that (1) the length of q is ℓ and (2) for any polynomial time algorithm A, the probability that A solves the CDH problem in $\mathbb{G}_1$ is negligible in ℓ.

3. Two Known CLAS Schemes

This section reviews Zhang-Zhang's CLAS scheme in [6] and Zhang-Qin-Wu-Zhang's CLAS scheme in [7].

3.1 Zhang-Zhang Scheme

Zhang-Zhang's CLAS scheme consists of following six algorithms. Let H_1, H_2, H_3 be cryptographic hash functions such that

$$H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1,$$
$$H_3 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{G}_1.$$

Setup(1^ℓ)**:** The algorithm chooses $\lambda \in Z_q^*$ randomly and sets $P_T = \lambda P$. Then it outputs λ as a master-key and P_T as a system parameter.

PartialPrivateKeyExtr(ID_i, λ)**:** This algorithm computes $Q_i = H_1(ID_i)$, $D_i = \lambda Q_i$ and outputs D_i as a partial private key for identity ID_i.

UserKeyGen(ID_i)**:** This algorithm chooses a random $x_i \in Z_q^*$ as user's secret key and sets $P_i = x_i P$ as his public key.

Sign$(ID_i, P_i, (x_i, D_i), M_i, \Delta)$**:** First, the algorithm chooses a random $r_i \in Z_q^*$ and computes

$$R_i = r_i P,$$
$$V = H_2(\Delta),$$
$$T_i = H_3(\Delta, M_i, ID_i, P_i, R_i),$$
$$S_i = D_i + x_i V + r_i T_i.$$

$\sigma_i = (R_i, S_i)$ is the output of this algorithm.

Aggr$((ID_1, \ldots, ID_n), (P_1, \ldots, P_n), (M_1, \ldots, M_n), (\sigma_1, \ldots, \sigma_n), \Delta)$: Let $\sigma_i = (R_i, S_i)$ for $1 \leq i \leq n$[†]. This algorithm computes

$$S = \sum_{i=1}^{n} S_i$$

and outputs $\sigma = (R_1, \ldots, R_n, S)$ as an aggregate signature.

Verify$((ID_1, \ldots, ID_n), (P_1, \ldots, P_n), (M_1, \ldots, M_n), \sigma, \Delta)$**:** Let $\sigma = (R_1, \ldots, R_n, S)$. This algorithm computes:

$$Q_i = H_1(ID_i), \quad \text{(for all } i\text{)}$$
$$V = H_2(\Delta),$$
$$T_i = H_3(\Delta, M_i, ID_i, P_i, R_i) \quad \text{(for all } i\text{)}$$

Then it checks if

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}\left(P_T, \sum_{i=1}^{n} Q_i\right) \hat{e}\left(V, \sum_{i=1}^{n} P_i\right) \prod_{i=1}^{n} \hat{e}(T_i, R_i).$$

If the equation holds, 1 (accept) is output, otherwise 0 (reject) is output.

In Zhang-Zhang scheme, the length of an aggregate signature is about half of (unprocessed) signatures. However, the length depends on the number of aggregated signatures.

If the CDH problem is difficult to solve, Zhang-Zhang scheme is secure against both type I and Type II attacks in the random oracle model.

3.2 Zhang-Qin-Wu-Zhang Scheme

Zhang-Qin-Wu-Zhang scheme is mostly similar to Zhang-Zhang scheme. This section only reviews the main distinct points. The scheme consists of following six algorithms. Let

$$H_1, H_2, H_3, H_4 : \{0,1\}^* \to \mathbb{G}_1,$$
$$H_5 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to Z_q^*$$

be hash functions.

Setup: mk is randomly chosen $\lambda \in Z_q^*$, $params$ is $P_T = \lambda P$.

PartialPrivateKeyExtr: This algorithm computes $((Q_{i,0}, Q_{i,1}) = (H_1(ID_i \| 0), H_1(ID_i \| 1))$. The partial private key is a pair $(D_{i,0}, D_{i,1}) = (\lambda Q_{i,0}, \lambda Q_{i,1})$.

UserKeyGen: The secret key is $x_i \in \mathbb{Z}_q^*$, the public key is $P_i = x_i P$.

Sign: Choose $r_i \in Z_q^*$ randomly, compute $R_i = r_i P$. Compute $T = H_2(\Delta)$, $V = H_3(\Delta)$, $W = H_4(\Delta)$, $h_i = H_5(M_i, \Delta, ID_i, P_i)$. Then compute

$$S_i = D_{i,0} + x_i V + h_i(D_{i,1} + x_i W) + r_i T.$$

Output $\sigma_i = (R_i, S_i)$ as a certificateless signature.

Aggr:

$$R = \sum_{i=1}^{n} R_i, \quad S = \sum_{i=1}^{n} S_i$$

are computed and $\sigma = (R, S)$ is output as the aggregate signature.

Verify: A verifier first computes $T = H_2(\Delta)$, $V = H_3(\Delta)$,

[†]all σ_i are assumed to be signed on the same state information Δ.

$W = H_4(\Delta)$. Then computes $Q_{i,0} = H_1(ID_i\|0), Q_{i,1} = H_1(ID_i\|1)), h_i = H_5(M_i, \Delta, ID_i, P_i)$ for all $i \in \{1, \ldots, n\}$. Finally verifies

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}\left(P_T, \sum_{i=1}^{n} Q_{i,0} + \sum_{i=1}^{n} h_i Q_{i,1}\right) \times \hat{e}\left(V, \sum_{i=1}^{n} P_i\right) \hat{e}\left(W, \sum_{i=1}^{n} h_i P_i\right) \hat{e}(T, R).$$

If it holds, output 1, otherwise output 0.

Zhang-Qin-Wu-Zhang scheme improves the length of aggregate signatures. However, each user cannot sign two or more messages with the same state information Δ. That is, the security of their scheme is not guarantee when a user signs on distinct messages M and M' by using the same state information Δ.

4. New Security Definition of CLAS

In [8], Shim proposed a new security requirement called *coalition resistance* by successfully presenting coalition attack on Zhang-Zhang scheme. Note that coalition of malicious KGC and users can happen when KGC itself acts fake users. Since existing two security definitions do not cover coalition resistance, here we present a new attack model (and a security definition) in which malicious KGC may corrupt some users. To strengthen the Type II adversary, we refer to the attack model, called malicious-but-passive KGC, defined for (non-aggregate) CLS by Wong et al. [9]. In Wong et al.'s model, KGC can not only corrupt some signers, but also deviate from SetUp algorithm. Based on their idea, our new attack model, called *strong malicioius KGC* (type III attack), has the following differences from type II attack.

- KGC can collude with some users (except a target user). That is, it can replace users public keys.
- KGC can generate *params* with a convenient manner for it.
- KGC can generate partial private key for some users with a convenient manner for it.

Before introducing the new attack model, we introduce a new notion, "Key-Verifiability."

Key-Verifiability: There exists an algorithm, KeyVerify, that taking ID_i and D_i (and *params*) as input, decides whether D_i is a valid partial private key for ID_i in polynomial time of security parameter.

Type III attack is modeled by using Game 3 described as follows. In this game, the adversary $\mathscr{A}_3$ includes a (strong) malicious KGC and malicious users, therefore it generates a master-key and *params* by itself, and also can carry out public-key replacements.

Setup: $\mathscr{C}$ sends security parameter 1^ℓ to $\mathscr{A}_3$. Then $\mathscr{A}_3$ chooses *params* at its best convenience and sends it to $\mathscr{C}$. Note that $\mathscr{A}_3$ does not have to follow **Setup** algorithm.

Attack: Adversary $\mathscr{A}_3$ is allowed to request the following types of queries in polynomial number of times:

- Create-User query $\mathcal{CU}(ID_i, D_i)$: With this query, $\mathscr{A}_3$ requests a public key of a user whose identity is ID_i. If identity ID_i has already been created, this request is rejected. Further, if the scheme has Key-Verifiability and D_i is not a valid partial private key for ID_i, this request is rejected. Otherwise, $\mathscr{C}$ runs UserKeyGen algorithm with input ID_i to obtain secret/public key pair (x_i, P_i), adds (ID_i, D_i, x_i, P_i) to the user list[†] and returns public key P_i to $\mathscr{A}_3$.
- Secret-Key query $\mathcal{SK}(ID_i)$: If the user with ID_i has been created, $\mathscr{C}$ outputs the secret key x_i stored in the user list.
- Public-Key-Replacement query $\mathcal{PKR}(ID_i, P'_i)$: If the user with ID_i has been created, $\mathscr{C}$ updates data in the user list from (ID_i, D_i, x_i, P_i) to $(ID_i, D_i, \perp, P'_i)$.
- Sign query $\mathcal{S}(ID_i, P_i, M_i, \Delta_i)$: If the user with ID_i has been created and the user's public key has not been replaced, $\mathscr{C}$ runs $\mathrm{Sign}(ID_i, P_i, (D_i, x_i), M_i, \Delta_i)$ to obtain a valid signature σ_i, then replies with σ_i.

Forgery: $\mathscr{A}_3$ outputs $(ID^*_1, \ldots, ID^*_n)$, $(P^*_1, \ldots, P^*_n)$, $(M^*_1, \ldots, M^*_n)$, Δ^*, and σ^*.

$\mathscr{A}_3$ wins Game 3 if:

1. $\mathrm{Verify}((ID^*_1, \ldots, ID^*_n), (P^*_1, \ldots, P^*_n), (M^*_1, \ldots, M^*_n), \sigma^*, \Delta^*) = 1$ holds, and
2. there exists at least one $i \in \{1, \ldots, n\}$ such that the user with ID^*_i has been created, $\mathscr{A}_3$ did not request ID^*_i's secret key nor replace ID^*_i's public key, and did not query $\mathcal{S}(ID^*_i, P^*_i, M^*_i, \Delta^*)$.

If the probability that $\mathscr{A}_3$ wins Game 3 is negligible for any polynomial time adversary $\mathscr{A}_3$, we say that the scheme is secure against type III attacks.

Clearly, security against type III attacks implies the security against type II attacks.

Shim's attack presented in [8] is a type III attack. That is, Zhang-Zhang scheme is not secure against type III attacks. On the other hand, for Zhang-Qin-Wu-Zhang scheme, we could not yet prove its security/insecurity against type III attacks.

5. New CLAS Schemes

In this section, we introduce two CLAS schemes which are provably secure against all of three types of attacks. The former one is similar to Zhang-Zhang scheme, while the latter one is similar to Zhang-Qin-Wu-Zhang scheme.

[†]The user list is initially empty.

5.1 The First Proposed CLAS Scheme: Scheme-1

This section now introduces our first proposed scheme: Scheme-1. Let

$$H_1 : \{0,1\}^* \to \mathbb{G}_1,$$
$$H_2 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{G}_1,$$
$$H_3 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{G}_1.$$

Setup algorithm, PartialPrivateKeyExtr algorithm, UserKeyGen algorithm, and Aggr algorithm are as same as those of Zhang-Zhang scheme. Note that Scheme-1, as well as Zhang-Zhang scheme, has Key-Verifiability; each user can check the validity of his partial private key D_i by verifying

$$\hat{e}(D_i, P) \stackrel{?}{=} \hat{e}(H_1(ID_i), P_T).$$

Sign: The user chooses a random $r_i \in Z_q^*$ and computes

$$R_i = r_i P,$$
$$V_i = H_2(\Delta, M_i, ID_i, P_i),$$
$$T_i = H_3(\Delta, M_i, ID_i, P_i, R_i),$$
$$S_i = D_i + x_i V_i + r_i T_i.$$

$\sigma_i = (R_i, S_i)$ is a certificateless signature on M_i.

Verify: To verify $\sigma = (R_1, \ldots, R_n, S)$, a verifier computes

$$Q_i = H_1(ID_i),$$
$$V_i = H_2(\Delta, M_i, ID_i, P_i),$$
$$T_i = H_3(\Delta, M_i, ID_i, P_i, R_i)$$

for all $1 \le i \le n$. Verify

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}\left(P_T, \sum_{i=1}^{n} Q_i\right) \prod_{i=1}^{n} \hat{e}(V_i, P_i) \prod_{i=1}^{n} \hat{e}(T_i, R_i).$$

If the equation holds, output 1, else output 0.

In Sect. 5.3, Scheme-1 is proved to be secure against 3 types of adversary. However, upon considering aggregate signature's length, it is inefficient as Zhang-Zhang scheme.

5.2 The Second CLAS Scheme: Scheme-2

The description of our second scheme, Scheme-2, is as follows. In this scheme, hash functions:

$$H_1, H_3 : \{0,1\}^* \to \mathbb{G}_1,$$
$$H_2 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{G}_1,$$
$$H_4 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to Z_q^*$$

are used.

SetUp, PartialPrivateKeyExtract, UserKeyGen: They are the same as those of Zhang-Qin-Wu-Zhang scheme. Scheme-2 also has Key-Verifiability; each user can check the validity of his partial private key $(D_{i,0}, D_{i,1})$ by verifying

$$\hat{e}(D_{i,0}, P) \stackrel{?}{=} \hat{e}(H_1(ID_i \| 0), P_T),$$
$$\hat{e}(D_{i,1}, P) \stackrel{?}{=} \hat{e}(H_1(ID_i \| 1), P_T).$$

Sign: To sign message M_i, $r_i \in \mathbb{Z}_q^*$ is chosen randomly. Then

$$R_i = r_i P,$$
$$V_i = H_2(\Delta, M_i, ID_i, P_i),$$
$$T = H_3(\Delta),$$
$$h_i = H_4(\Delta, M_i, ID_i, P_i),$$
$$S_i = D_{i,0} + h_i D_{i,1} + x_i V_i + r_i T.$$

Output $\sigma_i = (R_i, S_i)$ as a certificateless signature.

Aggr: As in Zhang-Qin-Wu-Zhang scheme, $\sigma = (R = \sum_{i=1}^{n} R_i, S = \sum_{i=1}^{n} S_i)$ is an aggregate signature.

Verify: To verify $\sigma = (R, S)$, a verifier computes

$$Q_{i,0} = H_1(ID_i \| 0),\ Q_{i,1} = H_1(ID_i \| 1),$$
$$V_i = H_2(\Delta, M_i, ID_i, P_i),$$
$$T = H_3(\Delta),$$
$$h_i = H_4(\Delta, M_i, ID_i, P_i)$$

for all i. Then, verify

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}\left(P_T, \sum_{i=1}^{n} (Q_{i,0} + h_i Q_{i,1})\right) \times \left(\prod_{i=1}^{n} \hat{e}(V_i, P_i)\right) \hat{e}(T, R).$$

If the equation holds, output 1, else output 0.

5.3 Security Analysis

In the random oracle model, our schemes are provably existentially unforgeable in the redefined security model of CLAS (3 types of adversary) under the CDH assumption. Formally, the following theorems can be shown.

Theorem 1: If there is no polynomial time algorithm that solves the CDH problem in group $\mathbb{G}_1$ with non-negligible probability, Scheme-1 is secure against type I attacks in the random oracle model.

Theorem 2: If there is no polynomial time algorithm that solves the CDH problem in group $\mathbb{G}_1$ with non-negligible probability, and if each user signs only one message related to Δ for any state information Δ, Scheme-2 is secure against type I attacks in the random oracle model.

We omit proofs of these theorems, since they are similar to Type I adversary's security proofs in [6] and [7], respectively

Theorem 3: If there is no polynomial time algorithm that solves the CDH problem in group $\mathbb{G}_1$ with non-negligible probability, Scheme-1 is secure against type III attacks in the random oracle model.

(Proof) We show that if there exists $\mathscr{A}_3$ that wins Game 3 of Scheme-1 with non-negligible probability, then we can construct a polynomially bounded algorithm $\mathscr{C}$ to solve the CDH problem with non-negligible probability. To this end, we show how $\mathscr{C}$, upon receiving a random instance (aP, bP), computes abP by interacting with $\mathscr{A}_3$ through Game 3.

Setup: At the beginning of Game 3, $\mathscr{C}$ sends security parameter 1^ℓ to $\mathscr{A}_3$. $\mathscr{A}_3$ outputs $params = P_T$ to $\mathscr{C}$. Three random oracles H_1, H_2 and H_3 are under $\mathscr{C}$'s control: Q-list for H_1, V-list for H_2, and T-list for H_3 are initially empty. In addition, $\mathscr{C}$ maintains User-list and S-List which are initially empty. Let q_{cu}, q_{sig} be the number of Create-User queries and Sign queries $\mathscr{A}_3$ can ask, respectively.

$\mathscr{C}$ randomly chooses $I \in [1, q_{cu}]$ in advance. Then, $\mathscr{C}$ answers $\mathscr{A}_3$'s various queries as follows. ($\mathscr{C}$ gives the same answer to formerly asked queries.)

Create-User queries: When $\mathscr{A}_3$ sends the i-th request $\mathcal{CU}(ID_i, D_i)$, $\mathscr{C}$ first checks that ID_i has not been created and D_i is valid. Next, $\mathscr{C}$ selects random $x_i \in \mathbb{Z}_q^*$. If $i = I$, $\mathscr{C}$ sets $ID_{\text{guess}} = ID_i$ and $P_i = x_i aP$, otherwise sets $P_i = x_i P$. Tuple $\langle ID_i; D_i, x_i, P_i\rangle$ is added to User-list and P_i is returned to $\mathscr{A}_3$.

Public-Key-Replacement queries: When $\mathscr{A}_3$ sends request $\mathcal{PKR}(ID_i, P_i')$, if $ID_i = ID_{\text{guess}}$, $\mathscr{C}$ aborts. Else, $\mathscr{C}$ updates User-list from $\langle ID_i; D_i, x_i, P_i\rangle$ to $\langle ID_i; D_i, \perp, P_i'\rangle$.

Secret-Key queries: When $\mathscr{A}_3$ sends request $\mathcal{SK}(ID_i)$, If $ID_i = ID_{\text{guess}}$, $\mathscr{C}$ aborts. Else, $\mathscr{C}$ retrieves a secret key x_i corresponding to ID_i from User-list, and returns x_i to $\mathscr{A}_3$.

H_1-hash queries: When $\mathscr{A}_3$ makes a query on $H_1(ID_i)$, $\mathscr{C}$ selects random $\phi \in \mathbb{Z}_q^*$, and computes $Q = \phi P$. Tuple $\langle ID_i; \phi, Q\rangle$ is added to Q-list and Q is returned to $\mathscr{A}_3$.

H_2-hash queries: When $\mathscr{A}_3$ makes a query on $H_2(\Delta, M_i, ID_i, P_i)$, $\mathscr{C}$ selects random $\pi \in \mathbb{Z}_q^*$. If $ID_i = ID_{\text{guess}}$, $\mathscr{C}$ sets $V = \pi bP$, otherwise sets $V = \pi P$. Tuple $\langle \Delta, M_i, ID_i, P_i; \pi, V\rangle$ is added to V-list, V is returned as a response.

H_3-hash queries: When $\mathscr{A}_3$ makes a query on $H_3(\Delta, M_i, ID_i, P_i, R_i)$, $\mathscr{C}$ selects random $\alpha \in \mathbb{Z}_q^*$, computes $T = \alpha P$, adds tuple $\langle \Delta, M_i, ID_i, P_i, R_i; \alpha, T\rangle$ to T-list, and returns T to $\mathscr{A}_3$.

Sign queries: $\mathscr{A}_3$ sends request $\mathcal{S}(ID_i, P_i, M_i, \Delta_i)$. $\mathscr{C}$ first searches ID_i in User-list and retrieves signing key (x, D). Further, $\mathscr{C}$ searches a tuple $\langle \Delta_i, M_i, ID_i, P_i; \pi, V\rangle$ in V-list for some (π, V). (If not available, H_2-hash query is issued internally.)

If $ID_i \neq ID_{\text{guess}}$, $\mathscr{C}$ can generate a signature $\sigma = (R, S)$ as Sign algorithm by using (x, D).

If $ID_i = ID_{\text{guess}}$, chooses random $r, \alpha \in \mathbb{Z}_q^*$ and sets

$$T = \alpha aP, \quad R = rP - \frac{x\pi}{\alpha} bP, \quad S = D + r\alpha aP.$$

If a tuple $\langle \Delta_i, M_i, ID_i, P_i, R; \alpha', T'\rangle$ is already in T-list for some (α', T'), then $\mathscr{C}$ aborts. (This event occurs with probability $1/q$, since R is randomly distributed in $\mathbb{G}_1$.) Otherwise, add $\langle \Delta_i, M_i, ID_i, P_i, R; \alpha, T\rangle$ to T-list.

Finally, $\mathscr{C}$ adds $\langle \Delta_i, M_i, ID_i, P_i; R, S\rangle$ to S-list and returns $\sigma = (R, S)$.

Forgery: $\mathscr{A}_3$ outputs $(ID_1^*, \ldots, ID_n^*), (P_1^*, \ldots, P_n^*)$, $(M_1^*, \ldots, M_n^*), \Delta^*$, and $\sigma^* = (R_1^*, \ldots, R_n^*, S^*)$. If $\mathscr{A}_3$ wins Game 3, the security model requires that σ^* is a valid aggregate signature and there exists ID_i^* among output n identities that (c1) a user whose identity is ID_i^* has been created, (c2) $\mathscr{A}_3$ neither requested this user's secret key nor replaced this user's public key, and (c3) query $\mathcal{S}(\Delta^*, M_i^*, ID_i^*, P_i^*)$ had not been submitted. If these requirements are not satisfied, $\mathscr{C}$ aborts. Further, the above ID_i^* is not the same as ID_{guess}, $\mathscr{C}$ aborts. Otherwise, $\mathscr{C}$ can compute abP as follows.

Without loss of generality, let $ID_{\text{guess}} = ID_1^*$. Since σ^* is valid, the verifying equation

$$\hat{e}(S^*, P) = \hat{e}\left(P_T, \sum_{i=1}^{n} Q_i^*\right) \prod_{i=1}^{n} \hat{e}(V_i^*, P_i^*) \prod_{i=1}^{n} \hat{e}(T_i^*, R_i^*)$$

holds, where

$$\begin{aligned} Q_i^* &= H_1(ID_i^*), \\ V_i^* &= H_2(\Delta^*, M_i^*, ID_i^*, P_i^*), \\ T_i^* &= H_3(\Delta^*, M_i^*, ID_i^*, P_i^*, R_i^*). \end{aligned}$$

By our setting, $\mathscr{C}$ knows $x_1^*, \pi_i^*, \phi_i^*, \alpha_i^*$ $(1 \le i \le n)$ that satisfy

$$\begin{aligned} P_1^* &= x_1^* aP, \quad V_i^* = \begin{cases} \pi_i^* bP & (i = 1) \\ \pi_i^* P & (i \neq 1), \end{cases} \\ Q_i^* &= \phi_i^* P, \quad T_i^* = \alpha_i^* P. \end{aligned}$$

($T_1^* = \alpha_1^* P$ comes from condition (c3).) Therefore, $\mathscr{C}$ can compute

$$abP = \frac{S^* - \sum_{i=1}^{n} \phi_i^* P_T - \sum_{i=2}^{n} \pi_i^* P_i^* - \sum_{i=1}^{n} \alpha_i^* R_i^*}{\pi_1^* x_1^*}.$$

Lastly, we show that $\mathscr{C}$ can compute the above abP with non-negligible probability.

From the above simulation, the simulation of $\mathscr{A}_3$'s environment is perfect until $\mathscr{C}$ aborts. This means that if $\mathscr{C}$ does not abort, $\mathscr{A}_3$ successfully forges an aggregate signature with non-negligible probability, and there exists ID_i^* satisfying the above conditions (c1),(c2),(c3).

From condition (c1), $ID_i^* = ID_{\text{guess}}$ holds with probability $1/q_{cu}$, If $ID_i^* = ID_{\text{guess}}$, $\mathscr{C}$ does not abort in any Secret-Key queries nor Public-Key-Replacement queries from condition (c2). Let define Abort as the event that $\mathscr{C}$ aborts as a result of any Sign queries. Then, the probability that $\mathscr{C}$ succeeds can be estimated as

$$\begin{aligned} &\Pr[\mathscr{C} \text{ succeeds}] \\ &= \frac{1}{q_{cu}} \Pr[\overline{\mathsf{Abort}}] \Pr[\mathscr{A}_3 \text{ wins Game 3}] \\ &\ge \frac{1}{q_{cu}} \left(1 - \frac{1}{q}\right)^{q_{sig}} \Pr[\mathscr{A}_3 \text{ wins Game 3}] \\ &\ge \frac{1}{q_{cu}} \left(1 - \frac{q_{sig}}{q}\right) \Pr[\mathscr{A}_3 \text{ wins Game 3}]. \end{aligned}$$

This means that non-negligible $\Pr[\mathscr{A}_3 \text{ wins Game 3}]$ implies non-negligible $\mathscr{C}$'s success probability. □

Theorem 4: If there is no polynomial time algorithm that solves the CDH problem in group $\mathbb{G}_1$ with non-negligible probability, and if each user signs only one message related to Δ for any state information Δ, Scheme-2 is secure against type III attacks in the random oracle model.

(Proof) The proof is similar to the proof of Theorem 3, therefore we only list out the essential points in the simulation. We consider H_1, H_2, H_3 as random oracles, while H_4 is an ordinary hash function. Let q_{cu}, q_{sig}, q_2 and q_3 be the number of Create-User queries, Sign queries, H_2-queries and H_3-queries, respectively. $\mathscr{C}$ randomly chooses $I \in [1, q_{cu}]$ and $J \in [1, q_2 + q_3 + q_{sig} + 1]$ in advance.

Create-User, **Public-Key-Replacement** and **Secret-Key queries** are simulated like in the proof of Theorem 3. Remember that $P_i = xaP$ if $ID_i = ID_{\text{guess}}$, $P_i = xP$ otherwise.

H_1-hash queries: If $\mathscr{A}_3$ requests $H_1(ID_i\|0)$ and $H_1(ID_i\|1)$, $\mathscr{C}$ chooses random $\phi_0, \phi_1 \in \mathbb{Z}_q^*$ and returns $Q_0 = \phi_0 P$ and $Q_1 = \phi_1 P$. $\langle ID_i; \phi_0, \phi_1, Q_0, Q_1\rangle$ is stored in Q-list.

H_2-hash queries: When $\mathscr{A}_3$ requests $H_2(\Delta, M_i, ID_i, P_i)$, $\mathscr{C}$ chooses a random $\pi \in \mathbb{Z}_q^*$. (If $H_3(\Delta_i)$ has not been queried, $\mathscr{C}$ makes such query internally.) If $ID_i \neq ID_{\text{guess}}$ or $\Delta_i \neq \Delta_{\text{guess}}$, $\mathscr{C}$ returns $V = \pi P$. Otherwise, $\mathscr{C}$ flips a coin c. If $c = 1$ (with probability 1/2), $\mathscr{C}$ returns $V = \pi P$. Otherwise, returns $V = \pi bP$. In any cases, $\langle \Delta_i, M_i, ID_i, P_i; c/\perp, \pi, V\rangle$ is added to V-list.

H_3-hash queries: To set a value of $H_3(\Delta_i)$, α is chosen randomly. In J-th query of H_3 queries, $\mathscr{C}$ sets $\Delta_{\text{guess}} = \Delta_i$ and $T = \alpha P$; otherwise, $T = \alpha aP$. T is the answer. $\langle \Delta_i; \alpha, T\rangle$ is added to T-list.

Sign queries: When $\mathcal{S}(\Delta_i, M_i, ID_i, P_i)$ is requested:[†]

If $ID_i \neq ID_{\text{guess}}$, $\mathscr{C}$ normally computes a signature.

If $ID_i = ID_{\text{guess}}$, first $\mathscr{C}$ searches User-list, V-list, and T-list, retrieves (D_0, D_1, x), (c, π), and α, respectively. Next $\mathscr{C}$ computes $h = H_4(\Delta_i, M_i, ID_i, P_i)$. Then, there are three cases: (If $H_3(\Delta_i)$ has not been queried, $\mathscr{C}$ makes such query internally.)

(case 1) If $\Delta_i = \Delta_{\text{guess}}$ and $c = 0$, $\mathscr{C}$ aborts.

(case 2) If $\Delta_i = \Delta_{\text{guess}}$ and $c \neq 1$, $\mathscr{C}$ computes

$$R = rP, \quad S = D_0 + hD_1 + \pi xaP + r\alpha P$$

for random $r \in \mathbb{Z}_q^*$.

(case 3) If $\Delta_i \neq \Delta_{\text{guess}}$, $\mathscr{C}$ chooses $r \in \mathbb{Z}_q^*$ randomly and computes

$$R = rP - \frac{x\pi}{\alpha}P, \quad S = D_0 + hD_1 + r\alpha aP.$$

In case 2 or 3, $\sigma = (R, S)$ is returned.

Forgery: $\mathscr{A}_3$ outputs $(ID_1^*, \ldots, ID_n^*)$, $(P_1^*, \ldots, P_n^*)$, $(M_1^*, \ldots, M_n^*)$, Δ^*, $\sigma^* = (R^*, S^*)$. If $\mathscr{A}_3$ does not win Game 3 or $\Delta^* \neq \Delta_{\text{guess}}$, $\mathscr{C}$ aborts. Otherwise, there exists an identity ID_i^* such that all conditions (c1),(c2),(c3) given in previous proof are satisfied. If $ID_i^* \neq ID_{\text{guess}}$ or coin c^* corresponding to $(\Delta^*, M_i^*, ID_i^*, P_i^*)$ is 1, $\mathscr{C}$ aborts. Finally, $\mathscr{C}$ computes abP as follows.

Without loss of generality, let $ID_1^* = ID_{\text{guess}}$. By our settings, $\mathscr{C}$ knows $\alpha^*, x_1^*, \pi_i^*, \phi_{i,0}^*, \phi_{i,1}^*$ $(1 \leq i \leq n)$ such that

$$T^* = \alpha^* P, \quad P_1^* = x_1^* aP$$
$$V_i^* = \begin{cases} \pi_i^* bP & (i = 1) \\ \pi_i^* P & (i \neq 1) \end{cases}$$
$$Q_{i,0}^* = \phi_{i,0}^* P, \quad Q_{i,1}^* = \phi_{i,1}^* P, \ (1 \leq i \leq n)$$

and

$$\hat{e}(S^*, P) = \hat{e}(P_T, \sum_{i=1}^{n}(Q_{i,0}^* + h_i Q_{i,1}^*)) \times \left(\prod_{i=1}^{n} \hat{e}(V_i^*, P_i^*)\right)\hat{e}(T^*, R^*),$$

where $h_i = H_4(\Delta^*, M_i^*, ID_i^*, P_i^*)$. Therefore, $\mathscr{C}$ can compute abP as

$$abP = \frac{S^* - \sum_{i=1}^{n}(\phi_{i,0}^* + h_i\phi_{i,1}^*)P_T - \sum_{i=2}^{n}\pi_i^* P_i^* - \alpha^* R^*}{\pi_1^* x_1^*}.$$

$ID_i^* = ID_{\text{guess}}$ and $\Delta^* = \Delta_{\text{guess}}$ hold with probability $1/q_{cu}(q_2 + q_3 + q_{sig} + 1)$[††] and in this case $\mathscr{C}$ does not abort in any Secret-Key queries and Public-Key-Replacement queries. If $ID_i^* = ID_{\text{guess}}$ and $\Delta^* = \Delta_{\text{guess}}$ hold, the probability that $\mathscr{C}$ aborts in Sign queries is at most 1/2, since $\mathscr{A}_3$ issues $\mathcal{S}(\Delta_{\text{guess}}, *, ID_{\text{guess}}, *)$ at most once from the assumption. $c^* = 0$ holds with probability 1/2. Therefore, we have

$$\Pr[\mathscr{C} \text{ succeeds}] \geq \frac{\Pr[\mathscr{A}_3 \text{ wins Game 3}]}{4q_{cu}(q_2 + q_3 + q_{sig} + 1)}.$$

□

6. Conclusion

In this paper, we present a new security definition for CLAS schemes in real-world situations and two new CLAS schemes. Our schemes have the highest security in the random oracle model assuming that the CDH problem is intractable. However, our schemes have trade-off; Scheme-1 is less efficient than Scheme-2, while Scheme-2 has a restriction that each user can not sign on two or more messages related to the same state information. (E.g., in Scheme-2, we cannot use the null string as a state information, and two signatures signed by a user cannot be aggregated.) Therefore, we may choose one of these schemes according to different purposes and requirements of communications.

An open problem is to construct a new CLAS scheme

[†]At this time, we can assume that the pair (Δ_i, ID_i) has not been queried to be signed from the assumption that each user signs only one message related to the same state information. If $\mathcal{S}(\Delta_i, M', ID_i, P')$ is requested after $\mathcal{S}(\Delta_i, M, ID_i, P)$ where $(M', P') \neq (M, P)$, then the latter request is rejected.

[††]Addition to q_3 times H_3-hash queries issued by $\mathscr{A}_3$, H_3-hash queries can be issued (at most) once for each H_2-query and Sign-query, and once for verification of the final forgery.

that achieves the same security level with our schemes, generates a constant size aggregate signature, and the state information can be used multiple times.

References

[1] S. Al-Riyami and G. Paterson, "Certificateless public key cryptography," ASIACRYPT 2003, Lect. Notes Comput. Sci. 2894, pp.452–473, 2003.
[2] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," EUROCRYPT 2003, Lect. Notes Comput. Sci. 2656, pp.416–432, 2003.
[3] C. Gentry and Z. Ramzan, "Identity-based aggregate signature," PKC 2006, Lect. Notes Comput. Sci. 3958, pp.257–273, 2006.
[4] M. Gorantla and A. Saxena, "An efficient certificateless signature scheme," CIS 2005, Part II, LNAI 3802, pp.110–116, 2005.
[5] Z. Gong, Y. Long, X. Hong, and K. Chen, "Two certificateless aggregate signatures from bilinear maps," SNPD 2007, Quingdao, IEEE Press, pp.188–193, 2007.
[6] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," Comput. Commun., vol.32, pp.1079–1085, 2009.
[7] L. Zhang, B. Qin, Q. Wu, and F. Zhang, "Efficient many-to-one authentication with certificateless aggregate signatures," Comput. Netw., vol.54, pp.2482–2491, 2010.
[8] K. Shim, "On the security of a certificateless aggregate signature scheme," IEEE Commun. Lett., vol.15, no.10, pp.1136–1138, 2011.
[9] S. Wong, H. Au, J. Chen, K. Liu, Y. Mu, and G. Yang, "Malicious KGC attacks in certificateless cryptography," ASIACCS'07, pp.20–22, March 2007.
[10] A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. Advances in Cryptology — CRYPTO'84, Lect. Notes Comput. Sci. 196, pp.47–53, 1985.

Nguyen Quoc Viet received his bachelor degree in computer engineering in 2013 from Tokyo Institute of Technology, Japan. He is currently pursuing Master's program in Engineering and Policy Analysis at Delft University of Technology, the Netherlands. His research interests include cryptography, information security and ICT management.

Wakaha Ogata received B.S., M.E. and D.E. degrees in electrical and electronic engineering in 1989, 1991 and 1994, respectively, from Tokyo Institute of Technology. From 1995 to 2000, she was an Assistant Professor at Himeji Institute of Technology. Since 2000 she has been working for Tokyo Institute of Technology, and now she is a Professor from 2013. Her current interests are cryptography and information security.